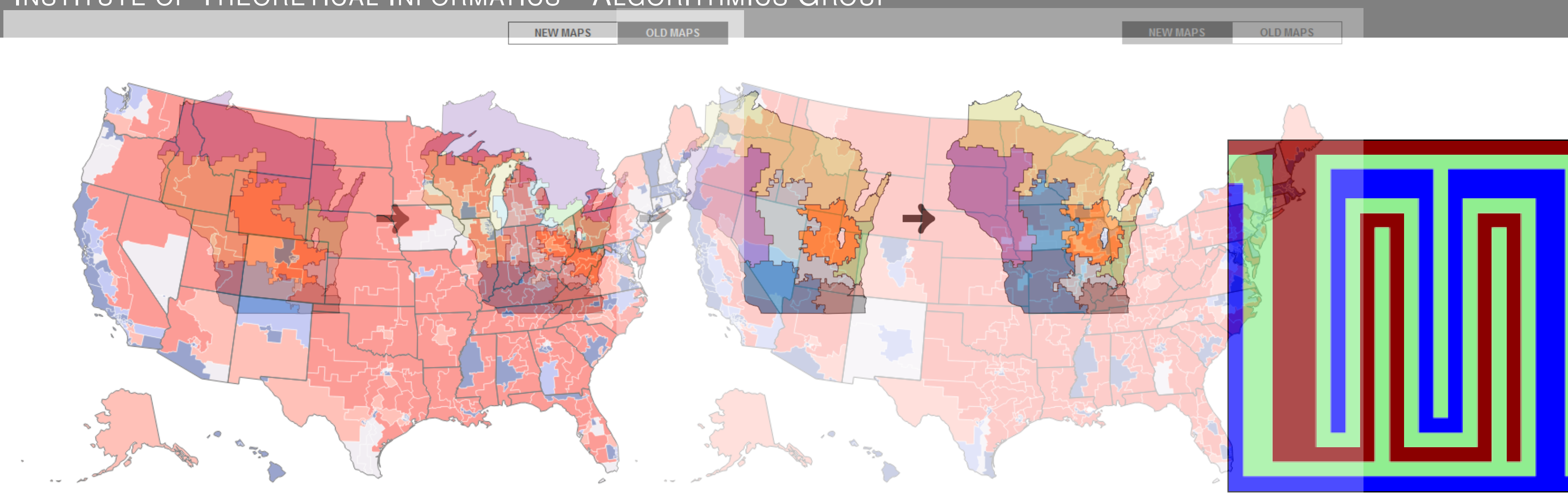
Karlsruhe Institute of Technology

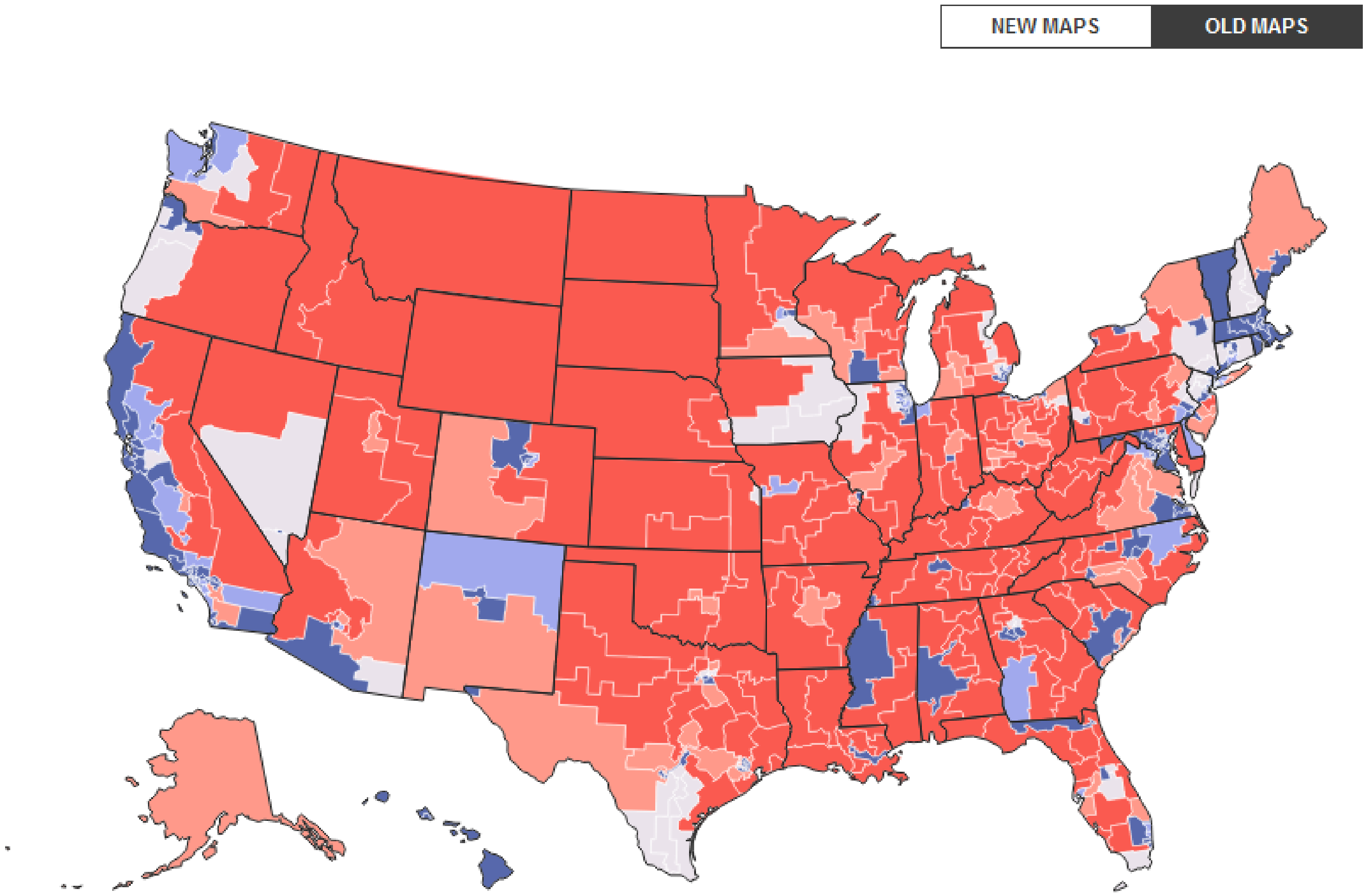
Reconfiguration of Polygonal Subdivisions via Recombination

Seminar Algorithmentechnik · December 15, 2023
Robert Külpmann

Institute of Theoretical Informatics · Algorithmics Group

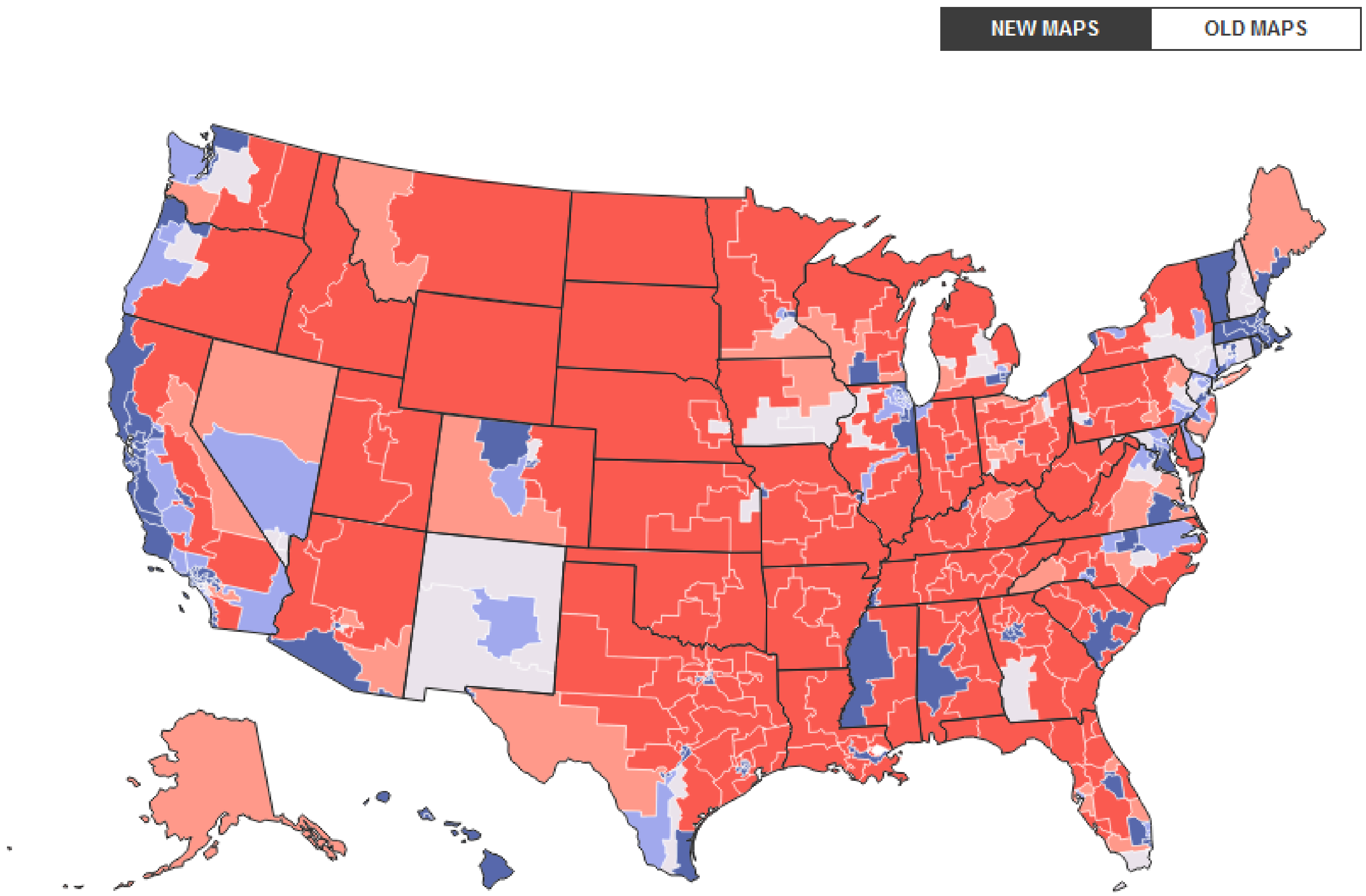


Redistricting USA



https://projects.fivethirtyeight.com/redistricting-2022-maps/
abgerufen 9.11.2023

Redistricting USA



https://projects.fivethirtyeight.com/redistricting-2022-maps/
abgerufen 9.11.2023

ReCom

Erfolgreichste Algorithmen für Redistricting
$\rightarrow$ Markovketten mit einer Reihe von Zustandsänderungen

Reombination Move (ReCom)

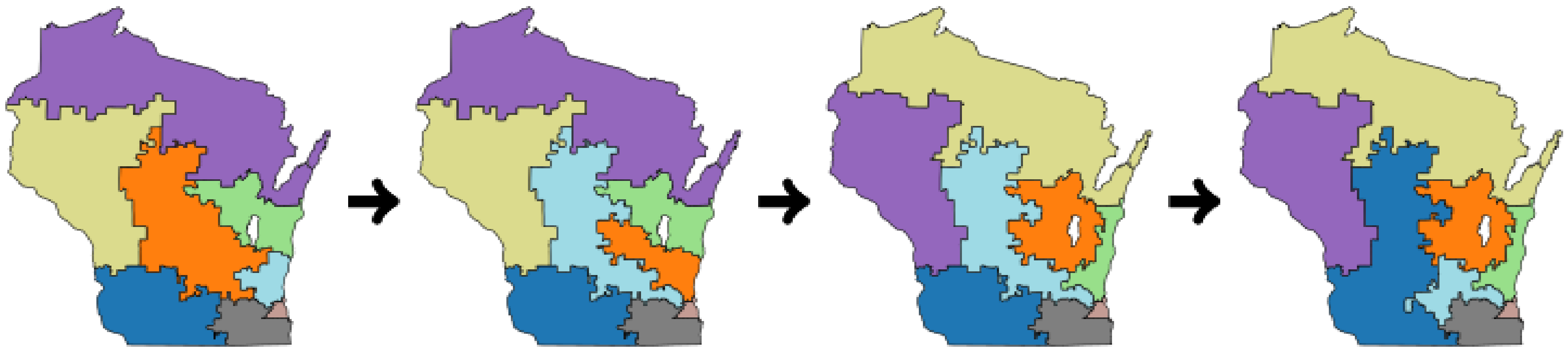
- Zwei benachbarte Bezirke
- Bewahrt Bevölkerungsgleichgewicht
- Bewahrt Verbindungen der Bezirke

ReCom

Erfolgreichste Algorithmen für Redistricting
$\rightarrow$ Markovketten mit einer Reihe von Zustandsänderungen

Reombination Move (ReCom)

- Zwei benachbarte Bezirke
- Bewahrt Bevölkerungsgleichgewicht
- Bewahrt Verbindungen der Bezirke



Ansatz

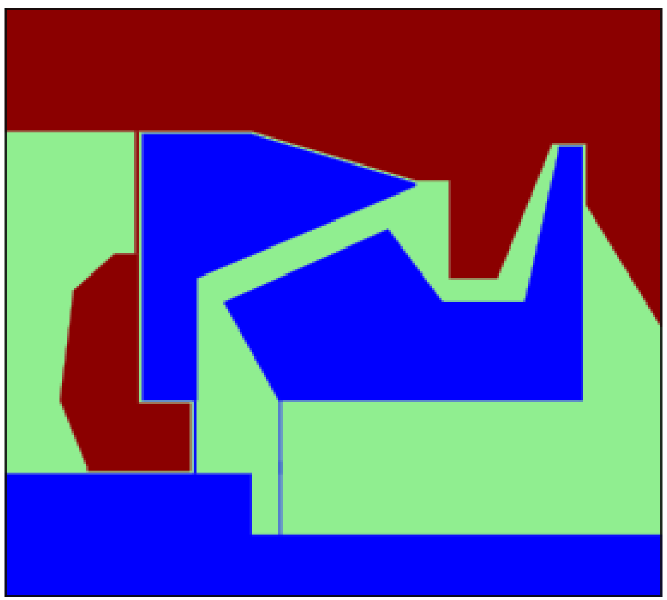
Statt

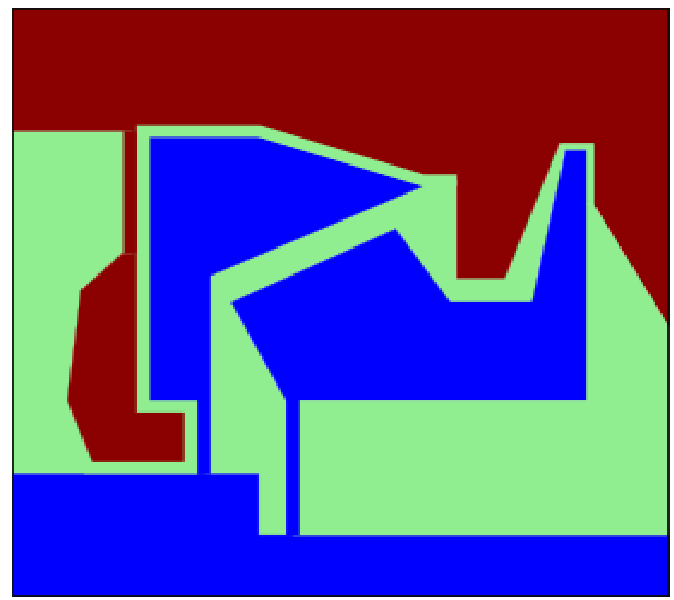
- Diskretes Modell $\rightarrow$ Kontinuierliches Modell
- Graphenproblem $\rightarrow$ Polygonale Partitionierung
- Nutzen einer "schwacher Repräsentation"

Ansatz

Statt

- Diskretes Modell $\rightarrow$ Kontinuierliches Modell
- Graphenproblem $\rightarrow$ Polygonale Partitionierung
- Nutzen einer ”schwacher Repräsentation”



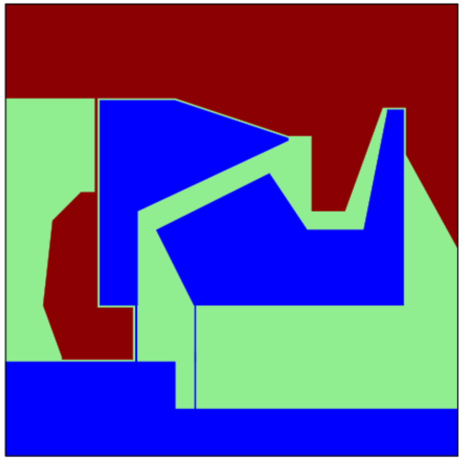


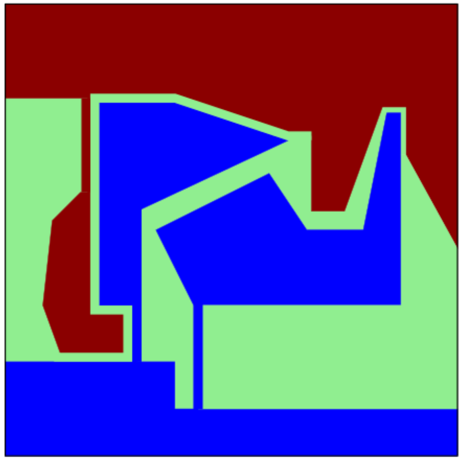
Schwache Repräsentation

Bezirke bleiben durch unendlich schmale Korridore verbunden.

Vorteile:

1. Korridore haben 0 Flächen
2. Anzahl der Knoten durch gemeinsame Knoten und überlappende Korridore wird verringert
3. Die Korridore können verbreitert werden, um eine Karte aus Polygonen zu erhalten





Konstruktion Einer ReCom Sequenz

Zwei Durchläufe:

1. Durchlauf auf einer schwachen Einbettung (Korridore mit 0 Fläche)
2. Durchlauf mit der Erweiterung auf Polygone

Voraussetzung:

- Zwei flächenkompatible *k* Gebietsmaps
- Eine flächenerhaltende ReCom Sequenz, mit Zwischenkarten mit schwachen Repräsentationen

ReCom Sequenz wird generiert, die $O(n)$ Knoten besitzt.

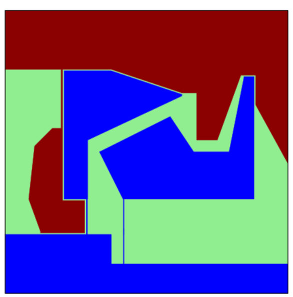
- flächenerhaltend: für alle i gilt, $Gebiet(D_i) = Gebiet(D_i)$
- flächenkompatibel: für alle i gilt $Gebiet(D_i) = Gebiet(\pi)$
  $\rightarrow$ mit Permutationen π

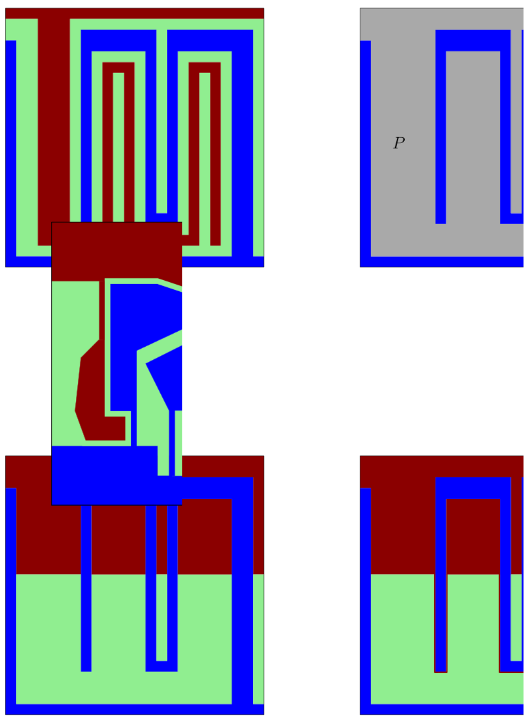
Institute of Theoretical Informatics
Algorithmics Group

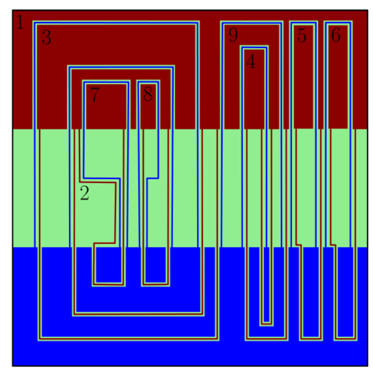
Rekonfigurationsalgorithmus für drei Bezirke

Algorithmus:

1. Vorverarbeitungsstufe: Bezirke sortieren - $O(1)$ Schritte
2. Gravitationsschritte: Bezirke in finale Position bringen - 3 Schritte
3. Austausch Sequenzen: Entfernen der Korridore - $O(logn)$ Schritte





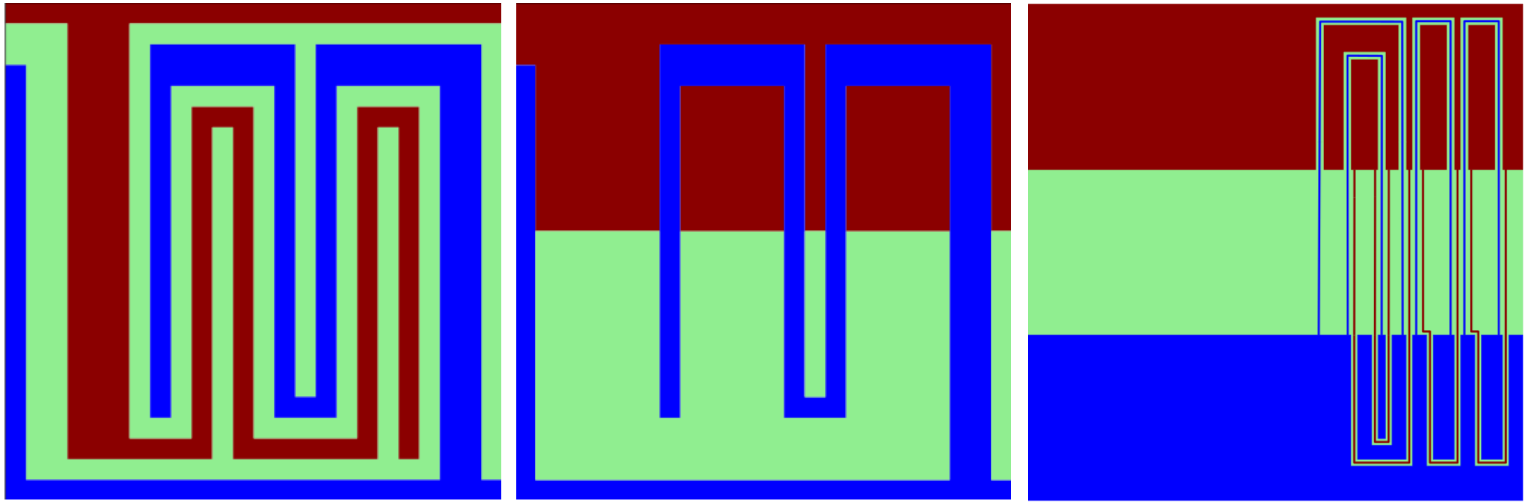


Institute of Theoretical Informatics
Algorithmics Group

Vorverarbeitungsstufe

Algorithmus:

1. Erstellen von Korridoren in $O(1)$ ReCom Schritten
2. Sortierung Top to Bottom
3. Sortiereigenschaft erstellen

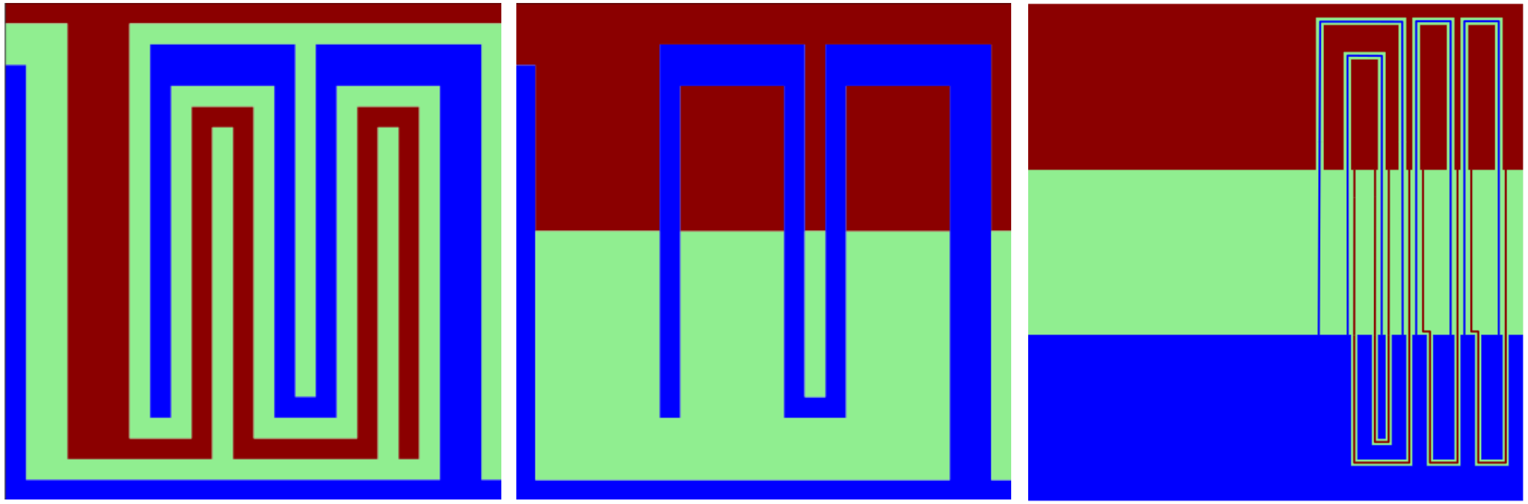


Institute of Theoretical Informatics
Algorithmics Group

Sortiereigenschaft

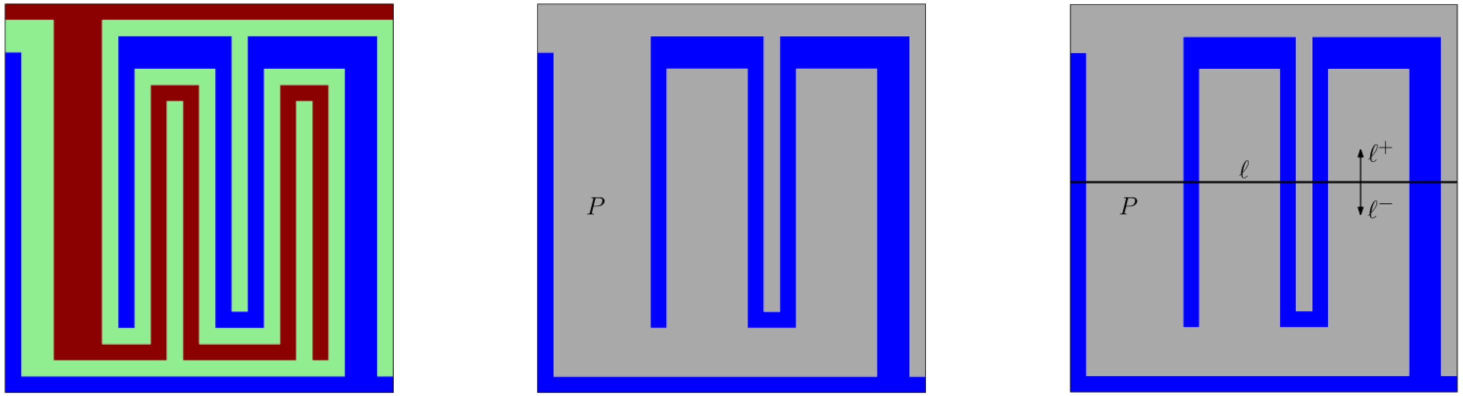
Definition:

Eine 3-Bezirkskarte erfüllt die Sortiereigenschaft, wenn der Schnittpunkt jedes Bezirks mit der rechten bzw. linken Seite der Domäne eine zusammenhängende Fläche bildet, und der mittlere Bezirk (Top to Bottom sortiert) eine Fläche besitzt, die größere oder gleich groß aller anderen Bezirke ist.

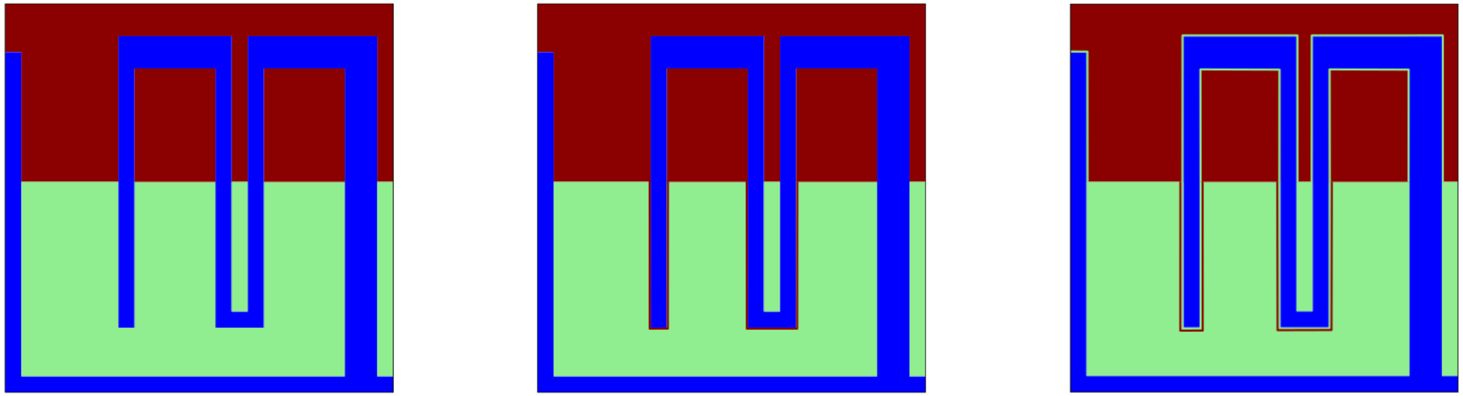


Gravitationsschritt

D_1 Rot
D_2 Blau
D_3 Grün

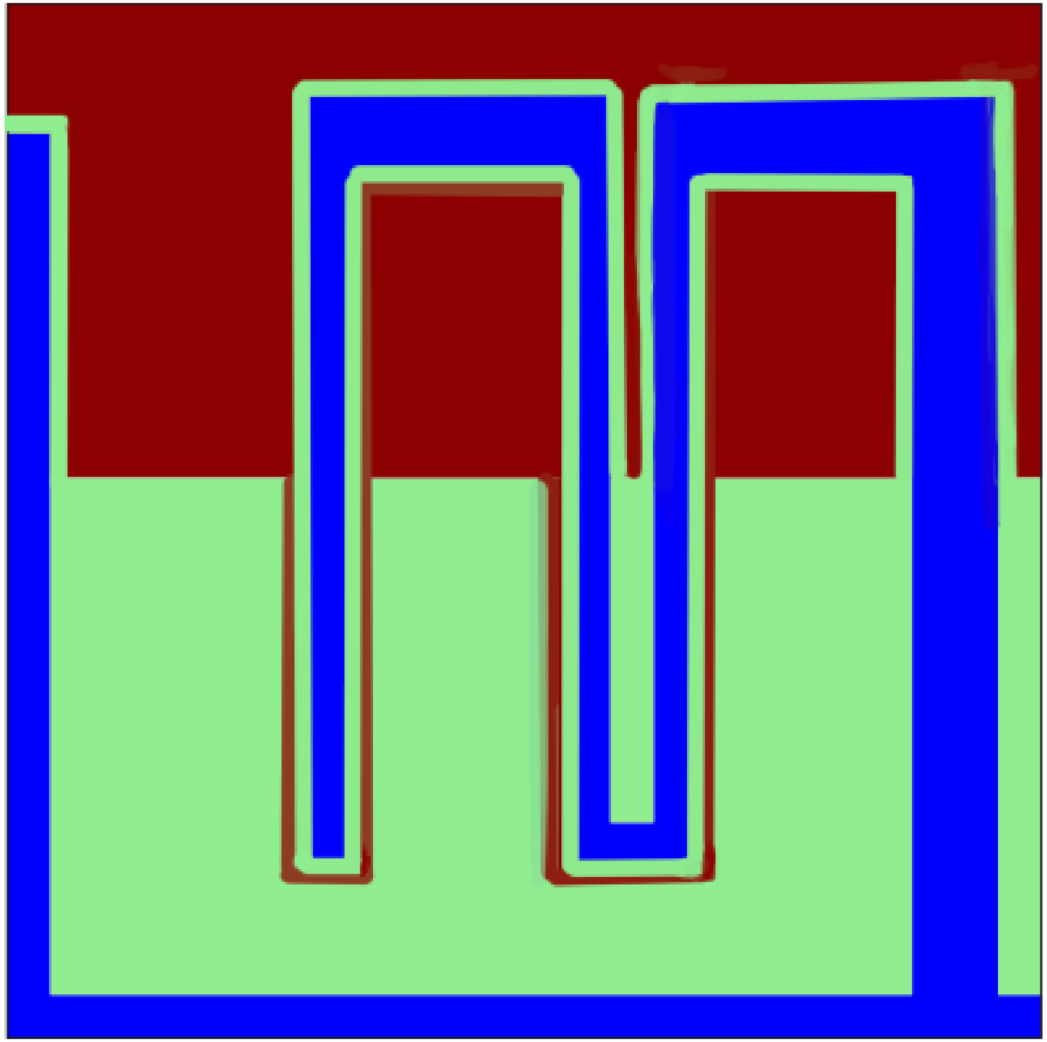


Erstellen der Korridore



Gravitationsschritt (2)

D_1 Rot
D_2 Blau
D_3 Grün



Erstellen der Korridore

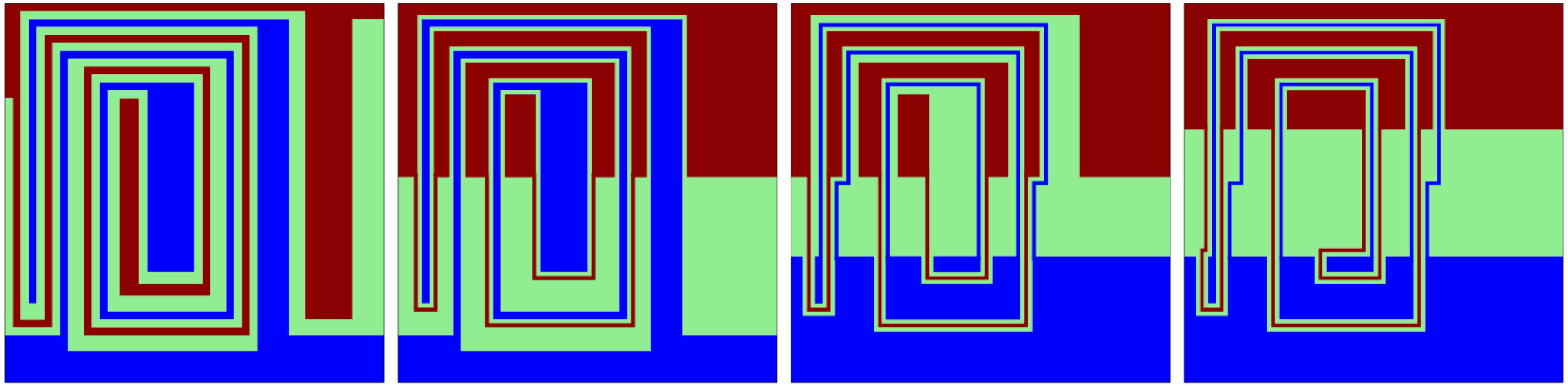
Institute of Theoretical Informatics
Algorithmics Group

Gravitationsschritt

Lemma

1. Annahme: D_1 und D_2 sind die obersten Bezirke,
2. Annahme: Die Bezirkskarte erfüllte die Sortierbedingung

Dann bezeichnet $GRAVITY(D_1, D_2)$ ein bezirkserhaltender ReCom Schritt.

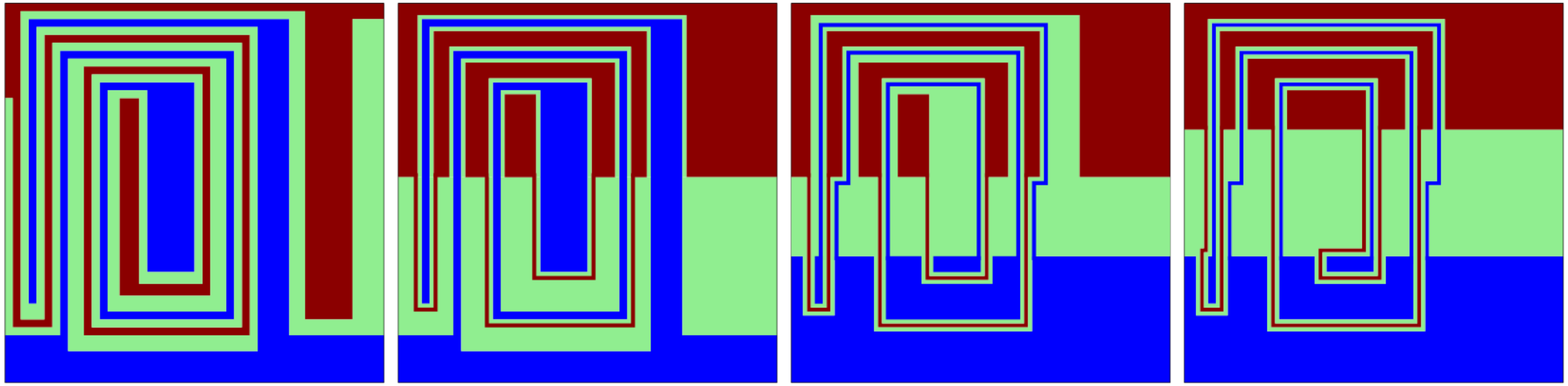


Gravitationsschritt (2)

Lemma

1. Annahme: Bezirkskarte M erfüllt die Sortierbedingung
2. D_1, D_2, D_3 Top to Bottom sortiert

Dann erzeugt $GRAVITY(D_1, D_2)$ eine Bezirkskarte, bei der Q_3, losgelöst von D_1' ist, mit ausnahme von Korridoren.



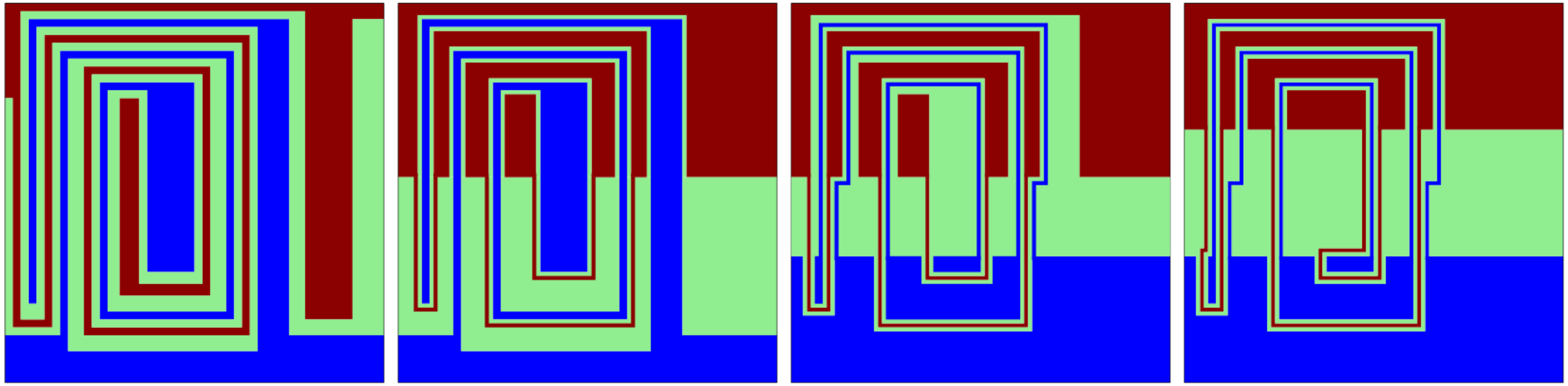
Institute of Theoretical Informatics
Algorithmics Group

Gravitationsschritt

Lemma

1. Annahme: Bezirkskarte M erfüllt die Sortierbedingung
2. D_1, D_2, D_3 Top to Bottom sortiert

Dann erzeugt die Sequenz $GRAVITY(D_1, D_2)$, $GRAVITY(D_2, D_3)$ $GRAVITY(D_1, D_2)$, eine Bezirkskarte, bei der D_1, D_2, D_3 sich in ihren eigenen kanonische Rechecke befinden, mit Ausnahme von Korridoren.



Regionen als Baum

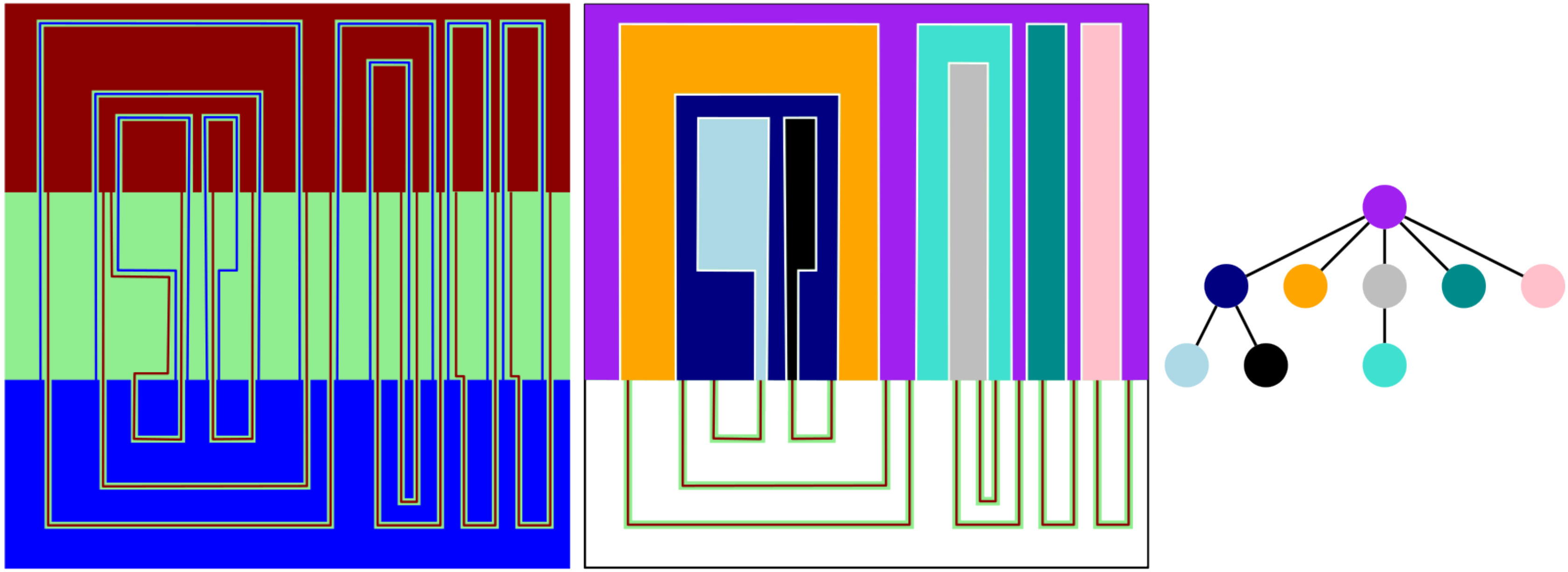
Ziel:

Definieren einer Baumrepräsentation zur Eliminierung von Korridoren

Betrachte das Rechteck D_1 D_2

Knoten: Polygone im Rechteck P

Kanten: Korridore in Q3 die Rechtecke in P verbinden

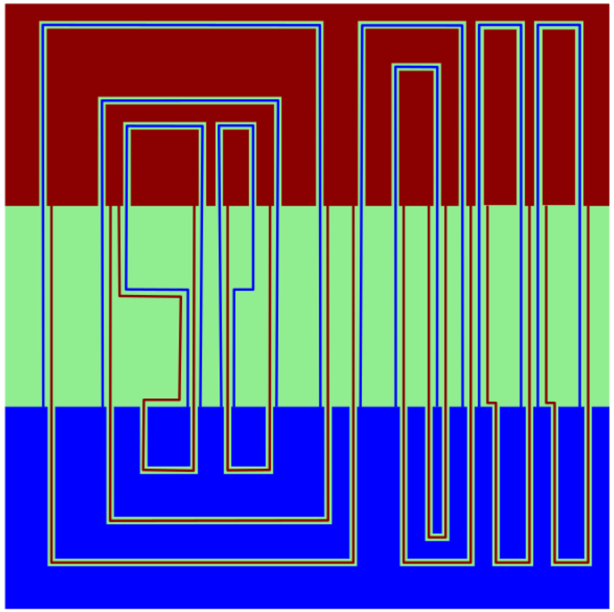


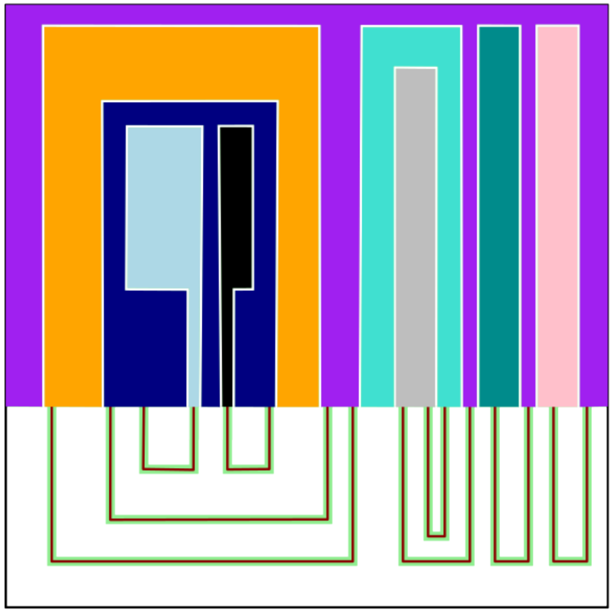
Institute of Theoretical Informatics
Algorithmics Group

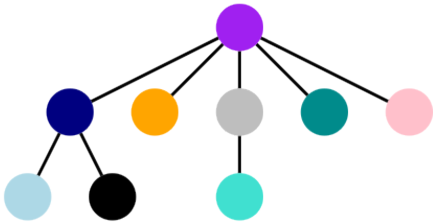
Lemma

Es existiert ein Subtree $T*$ von $T(P)$, so dass gilt:

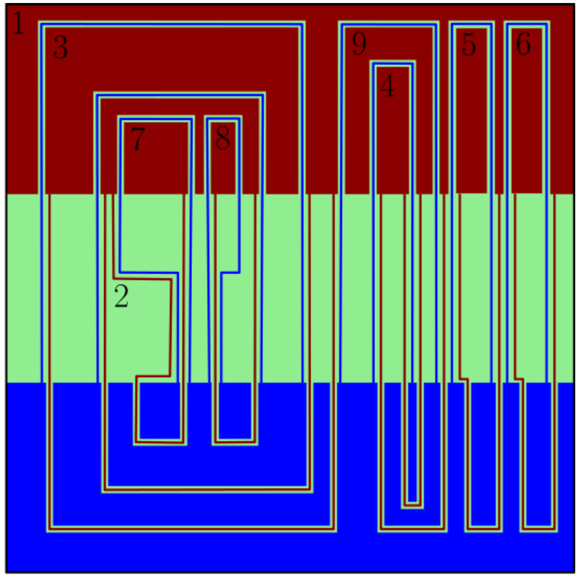
1. $T*$ beinhaltet mindestens $\frac{1}{3}$ der Knoten von $T(P)$
2. Das Gewicht von $T*$ ist maximal $\frac{W}{2} + w(c)$
3. dabei ist w(c) das Gewicht des Schwerpunktes c

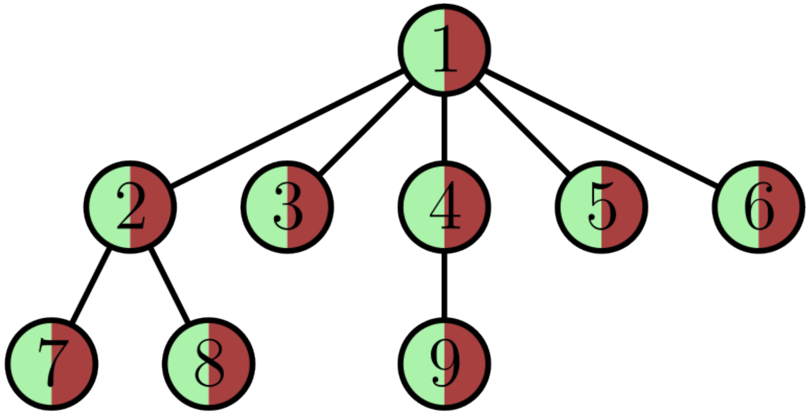




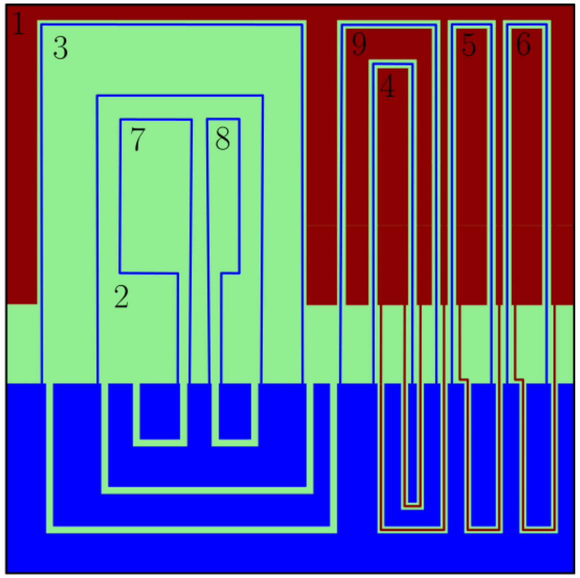


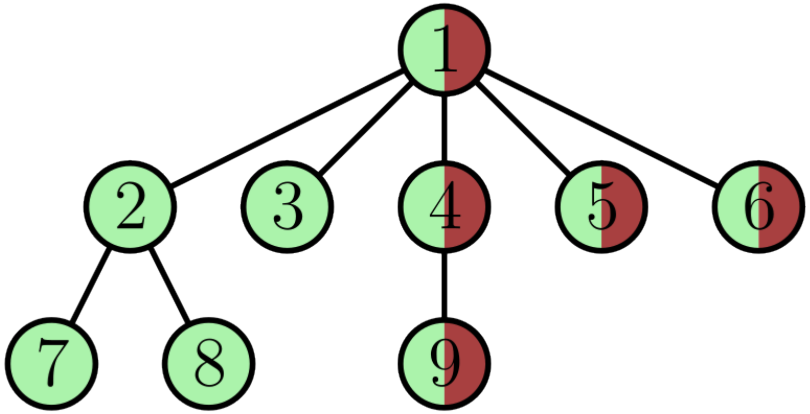
Austauschsequenz



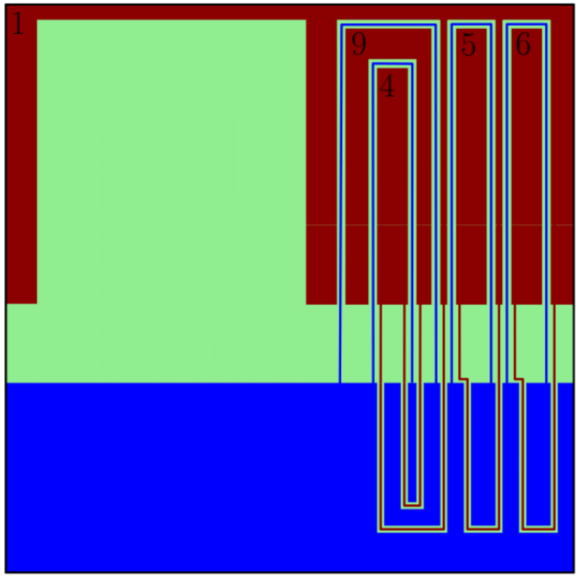


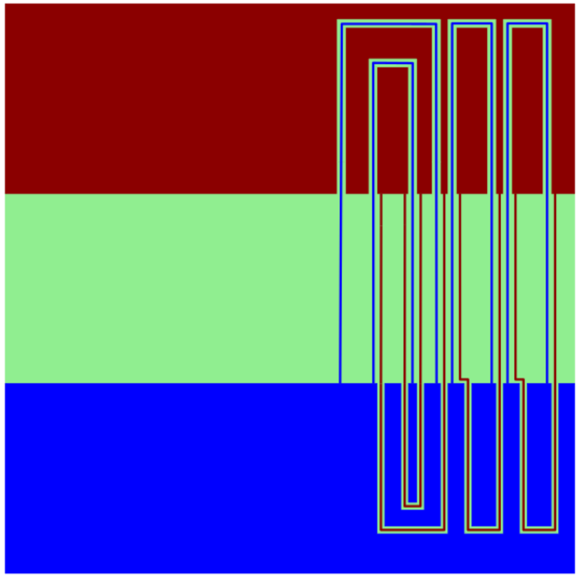
Austauschsequenz

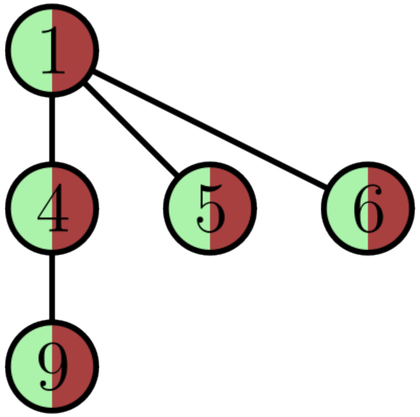




Austauschsequenz







Austauschsequenz

Theorem:
Für jede 3-Bezirkskarte mit Komplexität n, gibt es eine Sequenz von O(log n) ReCom Moves, welche diese in eine kanonische Karte transformiert.

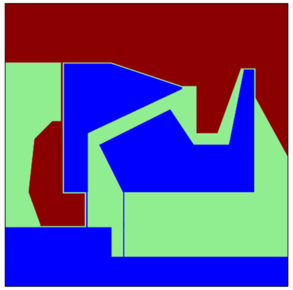
Drei Gravitationsschritte bringen eine Karte in kanonische Ordnung.
Jede Austauschsequenz eliminiert eine konstante Anzahl Korridore.

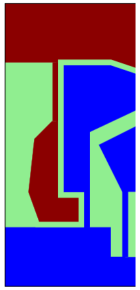
Institute of Theoretical Informatics
Algorithmics Group

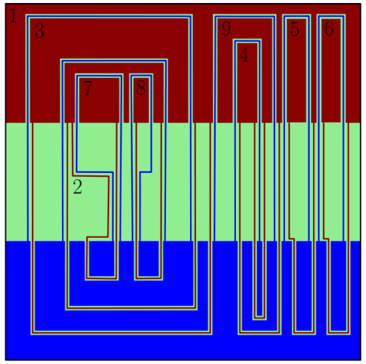
Rekonfigurationsalgorithmus für drei Bezirke

Algorithmus:

1. Vorverarbeitungsstufe: Bezirke sortieren - $O(1)$ Schritte
2. Gravitationsschritte: Bezirke in finale Position bringen - 3 Schritte
3. Austausch Sequenzen: Entfernen der Korridore - $O(log n)$ Schritte







Zusammenfassung

Ergebnisse:

- ReCom Schritte bei 3-Bezirkskarten: $O(\log n)$
- Zwischenkarten: Polygone mit O(n) Knoten

Weitere Ergebnisse des Papers:

- Lower Bound für 3-Bezirkskarten: $\Omega(\log n)$
- Obere Schranke für k-Bezirkskarten: $(\log n)^{O(\log k)}$
- Reduktion von k und n in Polynomialzeit möglich?